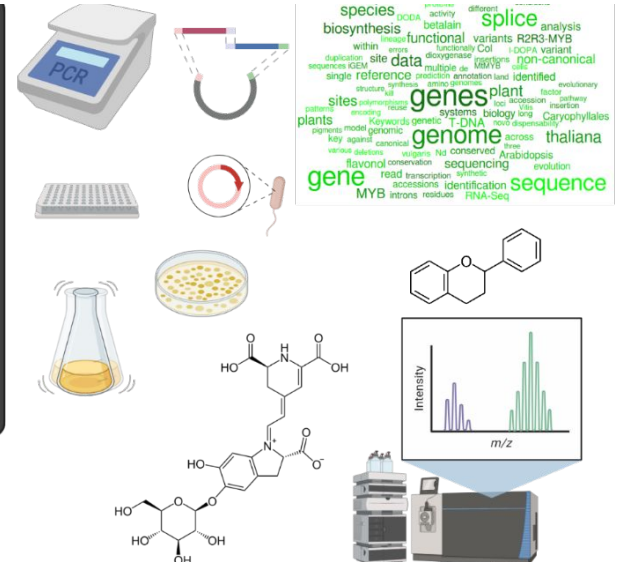
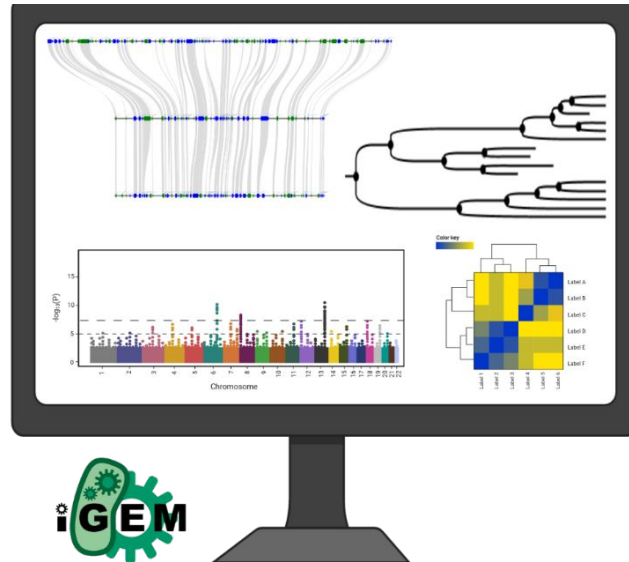
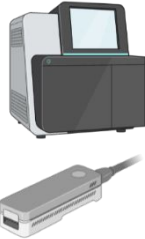




Technische
Universität
Braunschweig



Python - Modules

Prof. Dr. Boas Pucker (Plant Biotechnology and Bioinformatics)

Availability of slides

- All materials are freely available (CC BY) - after the lectures:
 - StudIP: 'Python for Life Scientists'
 - GitHub: <https://github.com/bpucker/teaching>
- Questions: Feel free to ask at any time
- Feedback, comments, or questions: [b.pucker\[a\]tu-bs.de](mailto:b.pucker[a]tu-bs.de)

My figures and content can be re-used in accordance with CC BY 4.0, but this might not apply to all images/logos. Some figure were constructed using bioRender.com.

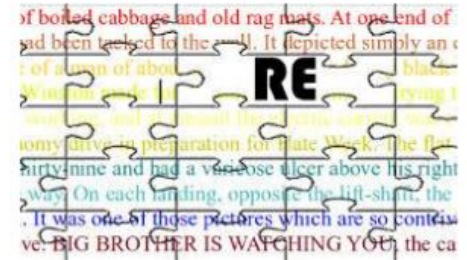
Concept of modules



NcbiblastnCommandline
NCBIXML



python™



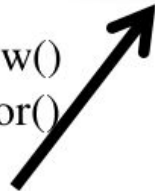
re.findall()
re.match()



np.array()



plt.show()
plt.color()



Importing modules

```
1  #basic import:
2  import re
3  #import of module under abbreviation:
4  import numpy as np
5  #import part of module:
6  from datetime import datetime
7
8  #usage of module functions:
9  re.findall()
10 datetime.now()
11
12 #try this (requires import):
13 print(str( datetime.now() ))
```

Run time calculation

- Current time is saved in two different places
- Difference is calculated to get the run time

```
1 from datetime import datetime
2
3 t1 = datetime.now()
4 #something should happen here
5 t2 = datetime.now()
6
7 print("it took " + str( t2-t1 ))
```

Regular expressions

- Regular expressions (=re) enable efficient search for substrings in a given string

```
1 import re
2 some_string = "AT2G12340.1|exon-1|23745-23965|AT2G12340.2exon-1_23745-23965"
3 hits = re.findall( "AT\dG\d{5}", some_string ) #generates list of hits
4 #searches for "AT\dG\d{5}"
5 #AT, G are matching the very same character
6 #\d is matching all number 0-9
7 #{5} specifies five repetitions of the previous element
8
9 print(hits)
```

Exercises - Part5

- 5.1) Write all AGIs of AtCol0_Exons.fasta into a new file!
- 5.2) Some IDs occur multiple times. Add a filter step to reduce the results to unique IDs!
- 5.3) Calculate frequency of each AGI and construct a histogram (matplotlib)!

Time for questions!